

REMARKS

Applicants respectfully request reconsideration of this application as amended. Claims 1-53 are pending in the application.

The Examiner rejected claims 1-2, 4-10, 21-33, 35-36, and 38-42 under 35 U.S.C. 103(a) as being unpatentable over "User Interface Markup Language (UIML) Draft Specification Document Version 17", Harmonia, Inc., in view of Ikemoto (US 5,969,717). Applicant respectfully disagrees.

Claim 1 of the present invention as claimed sets forth dynamically adapting a presentation in which a set of platform independent graphical user interface components are used to create a device platform dependent presentation by selectively transforming at least one graphical user interface component to adjust the size of the page to be closer to the maximum fill of a display screen of one of the heterogeneous device platforms running the application.

Neither Harmonia nor Ikemoto disclose the features discussed above with respect to Claim 1. As admitted by the Examiner, Harmonia does not disclose selectively transforming at least one graphical user interface component to adjust the size of the page to be closer to the maximum fill of a display screen of one of the heterogeneous device platforms running the application. Therefore, Harmonia does not disclose creating a device platform dependent presentation by selectively transforming at least one graphical user interface component to adjust the size of the page to be closer to the maximum fill of a display screen of one of the heterogeneous device platforms running the application.

The Examiner believes that Ikemoto discloses creating a device platform dependent presentation by selectively transforming at least one graphical user interface component to adjust the size of the page to be closer to the maximum fill of a display screen of one of the

heterogeneous device platforms running the application. Applicant respectfully submits that Ikemoto does not disclose such a teaching.

The Examiner cited column 11, lines 14-33 of Ikemoto. This portion of Ikemoto discusses positioning a window in the top left of the display screen with respect to top and left margins. Importantly, this window is not the entire display. Therefore, the sizing of this window does not adjust the maximum fill of a display screen. In response to Applicant's arguments, the Examiner pointed to column 11, lines 28-31 as teaching the operation of determining the maximum width and height of the display window. Upon review of this section, Applicant respectfully submits that these lines discuss the height and width of a display area of non-window items and the size of the window for "Visit Register". Visit Register is the name for all of the data items stored in the input data storage unit (col. 8, lines 40-41), not the overall display. Similarly, the non-window items do not constitute the entire display area. That is, individually or collectively, these do not constitute the entire display area of a display device. This section does not teach, mention, nor disclose adjusting to get closer to the maximum fill of a display screen of a heterogeneous device.

In the advisory action, the Examiner states that Ikemoto teaches that the "Visit Register" is included in a window of maximum width 3535 and maximum height of 3525 (see Ikemoto, column 11, lines 28-31). Further, Ikemoto teaches that the Graphical User Interface (GUI) builder generates code to product the display, which is then processed by the layout processing unit. This unit determines the size and display position of the Window (See Ikemoto, Column 11, lines 14-15). This unit then determines the size of the components, including both window and non-window components, to format the display based upon the size of the display window. Specifically, the component sizes are determined and calculated within the constructs of the total display size to determine the layout position of the entire display area (see Ikemoto, Column 11,

lines 16-67, and Column 12, lines 1-3). Figure 12 of Ikemoto shows the final display screen with the components formatted according to the above described processes.”

The Examiner raises the first point in his comments that states that the Visit Register is included in a window of maximum width 3535 and a maximum height of 3525. As stated above, the Visit Register is only one window on a display and these are non-window items that may be displayed. These would clearly be on the screen. Thus, the fact that one part of a display, i.e., the Visit Register window, has a maximum width and height is not the same as providing graphical user interface components that are arranged in a page as a function of hierarchy and then creating a presentation by transforming one or more of those graphical user interfaces to adjust the size of the page to be closer to the maximum fill of the display screen of one of a number of device platforms.

The Examiner seems to state that the GUI builder determines the size and display position of the window. The Examiner cites Column 11, lines 14-15 (not column 12, lines 28-31 as previously mentioned). The Examiner appears to consider the “window” referred to in this section as the entire display window. It absolutely is not. The window referred to in those sentences is the window referred to in the preceding paragraphs. Upon a review of column 10, starting at line 10 through the section referenced by the Examiner, it is clear that the display screen includes window and non window data items. By having window and non-window data items, it is clear that the positioning and sizing of a window is not the positioning and sizing of the whole display screen which, by the description here includes items that are not in the window specified by the Examiner.

In fact, Ikemoto discloses that the sizing of the window itself is done by adding a predetermined number of pixels to the height and width to the size of the display area of all the components that are included. In Ikemoto, a layout processing unit 5 calculates the position of

data items on a graphical user interface (GUI) display screen for which a GUI component has been selected and lays out the GUI component in a window. The layout processing unit determines the size and position of the GUI component allocated to each data item by adding pixel amounts to the size of the GUI component, typically by adding to the height and width of such a component. Once sized, the layout processing unit positions the window. Adding predetermined numbers of pixels to a window representing a portion of the display screen is clearly not the same as creating a device platform dependent presentation in which the size of the page is adjusted to be closer to the maximum fill of a display screen of one of the heterogeneous device platforms running the application.

Moreover, the Examiner states that the GUI builder formats a display based on the size of the display window. As we set forth above, that window is not the entire display and furthermore, the claim limitations are not solely directed to formatting a display based on size, but more specifically, and as specified in the claims, to adjust a size of a page that includes graphical user interface components in a hierarchical configuration to be closer to the maximum fill of a display screen of one of a device platform running on the application. We can easily envision formatting based on a display screen to make sure some position of a window or non-window data item is positioned with respect to some other component based on display screen, yet not be done to maximize the fill of the display screen.

As set forth in the claim language, the present invention arranges the graphical user components on a page as a function of hierarchy and thereafter creates a device platform dependent presentation by selecting GUI components to adjust the size of the page. Harmonia and Ikemoto do not teach, mention, nor disclose these limitations. Therefore, the combination of Harmonia and Ikemoto does not disclose all the limitations of Claim 1. Thus, the present invention as claimed is not obvious in view of their combination.

Similar limitations are also found in the other independent claims. In view of this, Applicant respectfully submits that the present invention as claimed is not obvious in view of Harmonia and Ikemoto.

Further, with respect to Claim 4, the claim sets forth:

‘The method of claim 1, wherein c) comprises selecting alternative graphical user interface components as a function of transformation rules when the display screen is over-filled by the page.’

As set forth in claim 4, it is clear that the claim requires selecting an alternative graphical user component as opposed to when the display screen is over-filled by the page. An alternative clearly implies that a graphical user interface (GUI) component that has been already provided in the hierarchical configuration, or that is already selected and arranged on a page, and another GUI component is substituted for it.

In the previous office action, the Examiner believes this is shown in Ikemoto, column 12, lines 39-54. This portion of Ikemoto is included below.

“A component selecting unit 4 is composed of a collation unit 41 and a decision unit 42. The collation unit 41 reads not only the various types of data items stored in the input data storage unit 2 but also the component selection rules from the selection rule storage unit 3, and selects a candidate for a suitable GUI component in accordance with the component selection rules for each of the data items stored in input data storage unit 2. Then, when a single screen input/output data item has a plurality of GUI component candidates sent from the collation unit 41, the decision unit 42 reads the component selection rules from the selection rule storage unit 3, and determines one GUI component candidate from the data characteristics held in the input data storage unit 2, and then stores the GUI component selection results for all the screen input/output data items in a component data storage unit 6.”

As set forth in the section specified by the Examiner, Applicant can find no such discussion of the switching of one GUI component that is already on a page with an alternative GUI component. It is clear that there are selection rules that selects a candidate for a suitable GUI component; however, there is no discussion of replacing one GUI component for another. This section only describes performing an initial selection to select GUI components based on a

set of rules. There is nothing that states that after selecting and determining that a display screen is overfilled, a new selection process is made and a replacement GUI component is selected. In view of this, Applicant respectfully submits that Claim 4 is not obvious in view of Harmonia and Ikemoto.

For the same reason as discussed with respect to Claim 4, Applicant respectfully submits that Claims 24 and 38 are not obvious in view of Harmonia and Ikemoto.

With respect to claim 5, the present invention as claimed sets forth the following:

“The method of claim 1, wherein c) comprises adding graphical user interface components to the page as a function of the hierarchical configuration when the display screen is under-filled by the page.”

As set forth in Claim 5, the claim requires adding a graphical user interface component to a page as a function of a hierarchical configuration when the display screen is under-filled by the page. The Examiner believes this is shown in Ikemoto at column 14, lines 39-61. The cited section of the patent has been included here below.

After the process at STEP 1401 has been completed, the GUI component selecting system reads the component selection rules, collates these with the uses for the inputted data items, and determines candidates for GUI components (STEP 1402). At this time, it is determined whether or not a plurality of GUI component candidates exist for one inputted data item (STEP 1403). If more than one candidate exists, a single specific candidate is selected on the basis of the feature of the data item, and the selection result is stored in the data storage unit 6 (STEP 1402).

Specifically, when the screen input/output data item entered from the data input unit 1 and its use and the degree of importance of the data item are stored in the input data storage unit 2, the GUI component selecting system sends an instruction to select a component to the collation unit 41 in the component selecting unit 4. The collation unit 41 reads not only the previously defined input data item from the input storage unit 2 but also the component selection rules used to cause the uses of data items inputted and outputted on a GUI display screen to correspond to GUI components on the basis of the selection rule storage unit 3. FIG. 17 is a table showing an example of the component selection rules stored in the selection rule storage unit 3.

However, the section above is silent with respect to selecting another component based on hierarchy when under-filled. Specifically, the section in Column 14, lines 39-48 does

mention determining candidates and does mention determining whether or not components candidates exist for a particular inputted data item and even includes selecting a specific candidate if more than one exists; however, there is nothing in this section that talks about adding a GUI component to a page as a function of the hierarchical configuration when a display screen is the page being under-filled by the page. There is no determination of the page being under-filled described in this section.

With respect to Column 14, lines 49-61, the first sentence only describes sending an instruction to select a component to the collation unit after the screen input/output data item has been entered and its use and degree of importance of the data item are stored. This sentence does not clearly imply or describe the selecting of a component based on a hierarchy when a display screen is under-filled by the page. The second sentence in the paragraph talks about reading input data items from storage and also the component selection rules used to cause the usage of the data item. Again, there is no discussion about selecting a GUI component based on the hierarchy when an under-filled page exists.

In view of the absence of a teaching of all the claims limitations, Applicant respectfully submits that the present invention as claimed in Claim 5 is not obvious in view of Harmonia and Ikemoto.

The same rationale applies to Claim 36, and therefore, Applicant respectfully submits Claim 36 is not obvious in view of these combinations.

The Examiner rejected claims 3, 11-20, 34, and 43-53 under 35 U.S.C. 103(a) as being unpatentable over "User Interface Markup Language (UIML) Draft Specification Document Version 17," Harmonia, Inc., in view of Ikemoto (US 5,969,717) as applied to claims 1 and 32 above, and further in view of Kashiwagi (US 6,037,939). As discussed above, Harmonia and Ikemoto do not disclose creating a device platform dependent presentation by selectively

transforming at least one graphical user interface component to adjust the size of the page to be closer to the maximum fill of a display screen of one of the heterogeneous device platforms running the application. Kashiwagi does not overcome that deficiency. Kashiwagi discloses an interaction tool that allows for interactive manipulation of data, not creating a device platform dependent presentation by selectively transforming at least one graphical user interface component to adjust the size of the page to be closer to the maximum fill of a display screen of one of the heterogeneous device platforms running the application. Therefore, the combination of Harmonia, Ikemoto and Kashiwagi does not disclose all the limitations of Claims 3, 11-20, 34, and 43-53. In view of this, Applicant respectfully submits that the present invention as claimed is not obvious in view of Harmonia, Ikemoto and Kashiwagi.

With respect to Claims 11, the same rationale above applies and is incorporated here by reference. Furthermore, the claim requires providing an intermediate representation comprising a plurality of container notes in a hierarchical configuration, identifying one of the container notes with the lowest hierarchical level and the highest layout priority in the intermediate representation, and arranging on a page at least one GUI component associated with the first container note. The Examiner seems to point to Harmonia, page 24, Section 6.4.1 as describing each of these three steps. However, Applicant can find no where in Section 6.4.1 that these three steps have been shown. Section 6.4.1 describes the “part” element. In Section 6.4.1, each part element corresponds to either one user interface widget or to nothing. It does state that parts may be nested to represent a hierarchical relationship of parts, including denoting one part nested inside of another part. However, there is no discussion of the intermediate representation being platform independent with respect to the plurality of the heterogeneous device platforms. Also, there is no discussion of identifying a first container with a lowest hierarchical level and the highest priority level in the intermediate representation and no discussion of arranging on a page

at least one graphical user interface component associated with first container note. Applicant requests the Examiner specify where he believes these are shown in Harmonia as they are not in Section 6.4.1. Applicant respectfully submits that it is Applicant's belief that those steps are not shown in Harmonia at all and, in view of that, Applicant respectfully submits that Claim 11 is not obvious in view of the combination of Harmonia, Ikemoto and Kashiwagi.

The same argument applies to Claims 43 and 53. Therefore, Applicants respectfully submit that Claims 43 and 53 are also not obvious in view of the combination of Harmonia, Ikemoto and Kashiwagi.

With respect to Claims 14 and 46, Claim 14 sets forth the following:

"The method of claim 11, wherein applying the transformation rule comprises: generating a list of possible graphical user interface components sorted by size; selecting a graphical user interface component from the list; and interchanging the at least one graphical user interface component arranged on the page in c) with the graphical user interface component from the list."

As set forth in Claim 14, the claim requires generating a list sorted by size and selecting a component from that list and interchanging it with one already arranged on a page. The Examiner sets forth that Ikemoto, Figure 9, shows a list of possible graphical user interface components sorted by size. Applicant respectfully requests the Examiner to explain where in Figure 9 there is an ordered list based on size. In column 704, 705, 706 and 707, none of the values are in numerical increasing or decreasing order. Therefore, they are not sorted by any size. The Examiner jumps back to column 2, lines 44-46 to set forth selecting a graphical user interface from a list however when interchanging a graphical user interface component that is arranged on a page with one from the list, the Examiner goes to column 14, lines 39-61. As discussed above, that portion of Ikemoto does not disclose interchanging a graphical user component. It deals with selecting graphical user components. Furthermore, it does not

interchange any graphical user interface component that is actually on a page. Applicant respectfully submits that the portions of Ikemoto selected by the Examiner does not teach, mention or disclose any of the elements in Claim 14. In view of this, Applicant respectfully submits that the invention as claimed is not obvious in view of the combination of Harmonia, Ikemoto and Kashiwagi.

The Examiner rejected claim 37 under 35 U.S.C. 103(a) as being unpatentable over "User Interface Markup Language (UIML) Draft Specification Document Version 17," Harmonia, Inc., in view of Ikemoto (US 5,969,717) as applied to claim 32 above, and further in view of Orr (US 5,895,477). As discussed above, Harmonia and Ikemoto do not disclose creating a device platform dependent presentation by selectively transforming at least one graphical user interface component to adjust the size of the page to be closer to the maximum fill of a display screen of one of the heterogeneous device platforms running the application. Orr does not overcome that deficiency. Orr discloses dropping content objects onto a composition, not creating a device platform dependent presentation by selectively transforming at least one graphical user interface component to adjust the size of the page to be closer to the maximum fill of a display screen of one of the heterogeneous device platforms running the application. Therefore, the combination of Harmonia, Ikemoto and Orr does not disclose all the limitations of Claim 37. In view of this, Applicant respectfully submits that the present invention as claimed is not obvious in view of Harmonia, Ikemoto and Orr.

Accordingly, Applicant respectfully submits that the objections to the claims and the abstract have been overcome by the amendments and the remarks and withdrawal of these rejections is respectfully requested. Applicant submits that Claims 1-53 as amended are in condition for allowance and such action is earnestly solicited.

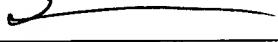
Please charge any shortages and credit any overcharges to our Deposit Account No.

02-2666.

Respectfully submitted,

BLAKELY SOKOLOFF TAYLOR & ZAFMAN LLP

Dated: 2/22, 2006



Michael J. Mallie
Attorney for Applicant
Registration No. 36,591

12400 Wilshire Boulevard
Seventh Floor
Los Angeles, CA 90025-1026
(408) 720-8598